

RAPPORT PROJET : FITPLANNER

Membres : Erwin DZIUBA & Gwenolé BINET



SOMMAIRE

Introduction	3
Objectifs du projet	3
Fonctionnalités	4
Contraintes Techniques	4
Documentation technique et utilisateur	5
Gestion de projet	13
Conclusion.....	15
Code source.....	16
Ressources supplémentaires et références	17

INTRODUCTION

La musculation fait partie des sports les plus pratiqué en France. Une séance de musculation est composée de plus exercices et un exercice est composé de plusieurs séries et de répétitions. Il y a des personnes qui vont utiliser un carnet papier, d'autres se souviendront de mémoire quel séance faire, quel jour, combien d'exercices, quel muscle etc.



C'est pourquoi nous avons décider de créer un planificateur pour les personnes qui pratiquent la musculation en version numérique et plus simple et rapide d'utilisation qu'un carnet, et plus fiable que la mémoire !

OBJECTIFS DU PROJET

Notre objectif est de créer une application permettant à l'utilisateur de planifier ses séances de musculation. L'utilisation sera personnalisée en fonction des besoins de l'utilisateur. L'utilisateur disposera d'un compte personnel, sur lequel il pourra enregistrer ses séances. Pour planifier une séance, l'utilisateur devra, choisir un titre à sa séance, la durée, la date, une description, le muscle qu'il voudra travailler et l'exercice associé au muscle. Cela sera accompagné d'une interface graphique intuitive qui garantira une expérience utilisateur ergonomique.

Fonctionnalités

Les fonctionnalités principales de notre application :

- L'utilisateur peut se connecter
- L'utilisateur peut se créer un compte
- Possibilité de créer une séance grâce à la saisie de : titre de la séance, la durée (en min), la date (format année/mois/jours), sélectionner le muscle qu'on veut entraîner et sélectionner l'exercice qu'on veut faire associé à ce muscle
- Possibilité d'avoir plusieurs exercices dans une séance
- Possibilité d'enregistrer plusieurs séances

Contraintes Techniques

Plusieurs contraintes techniques sont à respecter lors de la réalisation du projet :

- L'application doit être écrite en flutter (dart)
- Le succès du projet (l'application fait ce qu'elle est censée faire).
- La qualité de l'IHM et de l'expérience utilisateur, la portabilité de l'interface (adaptation aux différents écrans...).
- L'utilisation d'une API ou plusieurs API.
- La qualité du code et de sa structure. Il est attendu que le code soit commenté (les commentaires ne sont pas comptabilisés dans le nombre de lignes). La qualité du code implique un très faible recours au copier-coller et une factorisation importante du code produit.
- Le sujet du projet portera sur l'utilisation d'une API ou plusieurs API avec Flutter .
- Le projet doit permettre de produire au moins 1000 lignes de code validées (c'est-à-dire testées) et doit être réalisé sous Flutter, en binôme.

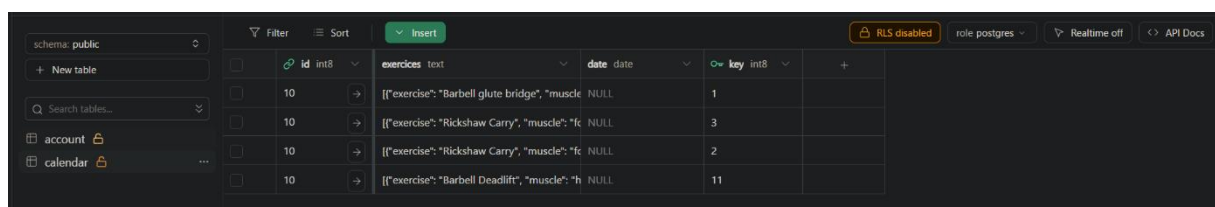
Documentation technique et utilisateur

Développement et outils utilisé :

Nous avons utilisé **Android Studio** pour développer cette application web.

L'API que nous avons utilisée est : **API NINJAS**

Pour la gestion des données, nous avons opter pour **Supabase**

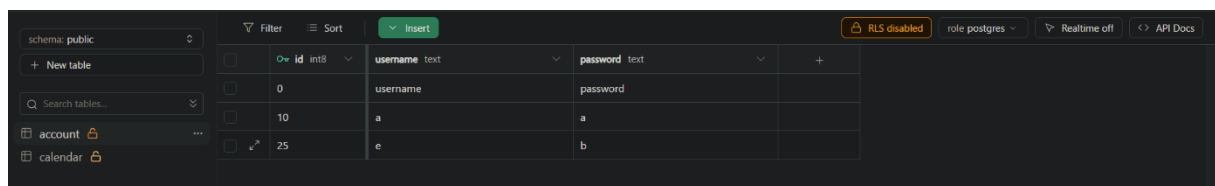


Schema: public

Filter Sort Insert

RLS disabled role: postgres Realtime off API Docs

	id int8	exercices text	date date	key int8	
	10	[{"exercise": "Barbell glute bridge", "muscle": "glutes"}]	NULL	1	
	10	[{"exercise": "Rickshaw Carry", "muscle": "feet"}]	NULL	3	
	10	[{"exercise": "Rickshaw Carry", "muscle": "feet"}]	NULL	2	
	10	[{"exercise": "Barbell Deadlift", "muscle": "hamstrings"}]	NULL	11	

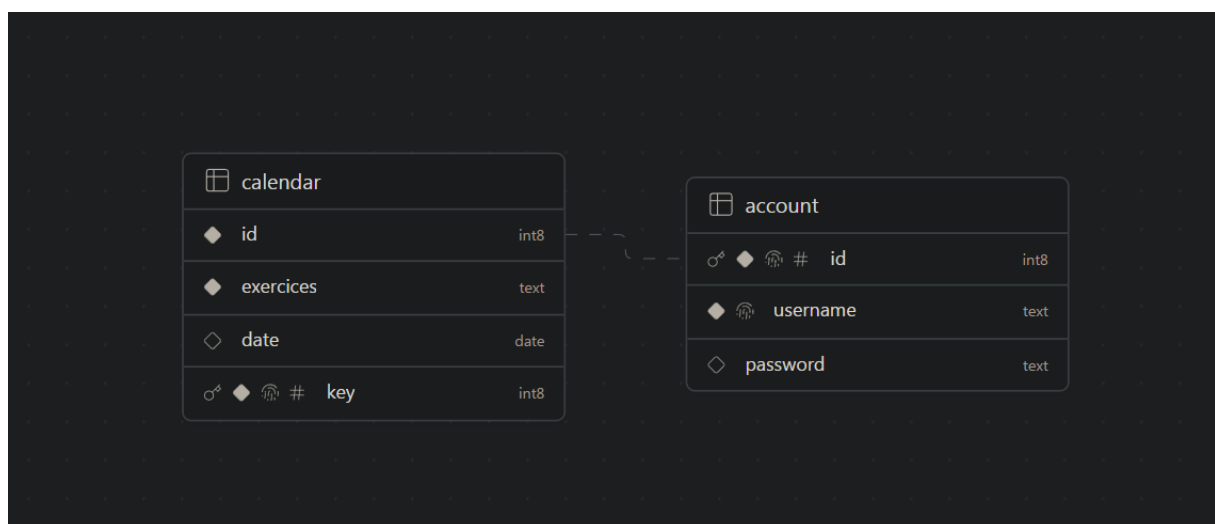


Schema: public

Filter Sort Insert

RLS disabled role: postgres Realtime off API Docs

	id int8	username text	password text	
	0	username	password	
	10	a	a	
	25	e	b	



Fonctionnement de la Base de Données du Projet FITPLANNER

La base de données du projet FITPLANNER est conçue pour stocker et organiser les informations essentielles à l'application de planification de séances de musculation. Elle est structurée en deux tables principales qui assurent le stockage des données relatives aux utilisateurs et à leurs séances d'entraînement.

1. Table account :

Cette table est au cœur de la gestion des utilisateurs. Elle stocke les données d'identification et d'authentification, permettant ainsi aux utilisateurs de créer un compte et de se connecter à l'application. Chaque ligne représente un compte utilisateur distinct avec les champs suivants :

id (type int8) : Il s'agit d'une clé primaire qui assigne un numéro d'identification unique et automatique à chaque utilisateur pour garantir l'unicité.

username (type texte) : Ce champ stocke le nom d'utilisateur, choisi par l'utilisateur au moment de la création de son compte.

password (type texte) : Le mot de passe est également enregistré dans ce champ. Il est à noter que dans un contexte de production, ce mot de passe devrait être chiffré pour des raisons de sécurité.

2. Table calendar :

La table calendar est utilisée pour enregistrer les séances de musculation planifiées par les utilisateurs. Chaque enregistrement contient les détails d'une séance spécifique, répartis dans les colonnes suivantes :

id (type int8) : Similaire à la table account, ce champ est une clé primaire qui attribue un identifiant unique à chaque séance pour éviter les doublons.

exercices (type texte) : Ce champ est particulièrement important puisqu'il contient une chaîne de texte formatée en JSON, qui liste les exercices choisis par l'utilisateur, ainsi que les muscles ciblés par ces exercices.

date (type date) : Indique la date prévue pour la séance, permettant à l'utilisateur de planifier ses entraînements.

key (type int8) : Ce champ est vraisemblablement une clé étrangère qui fait le lien entre une séance et un compte utilisateur de la table account, établissant ainsi une relation entre l'utilisateur et ses séances d'entraînement planifiées.

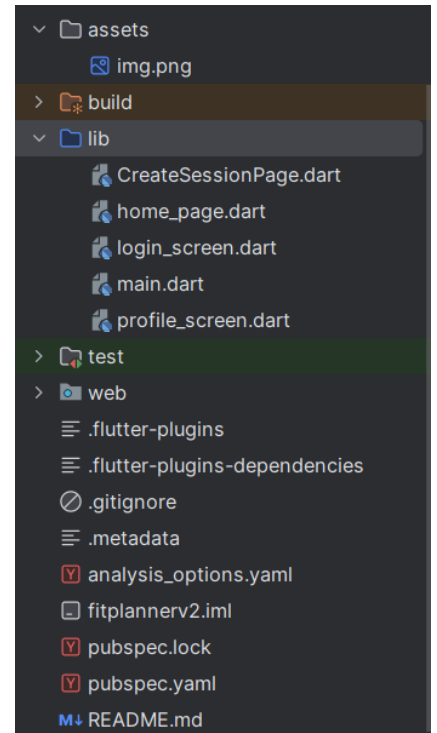
L'architecture de notre code :

Assets :

- Img.png (image utilisé pour le profile screen)

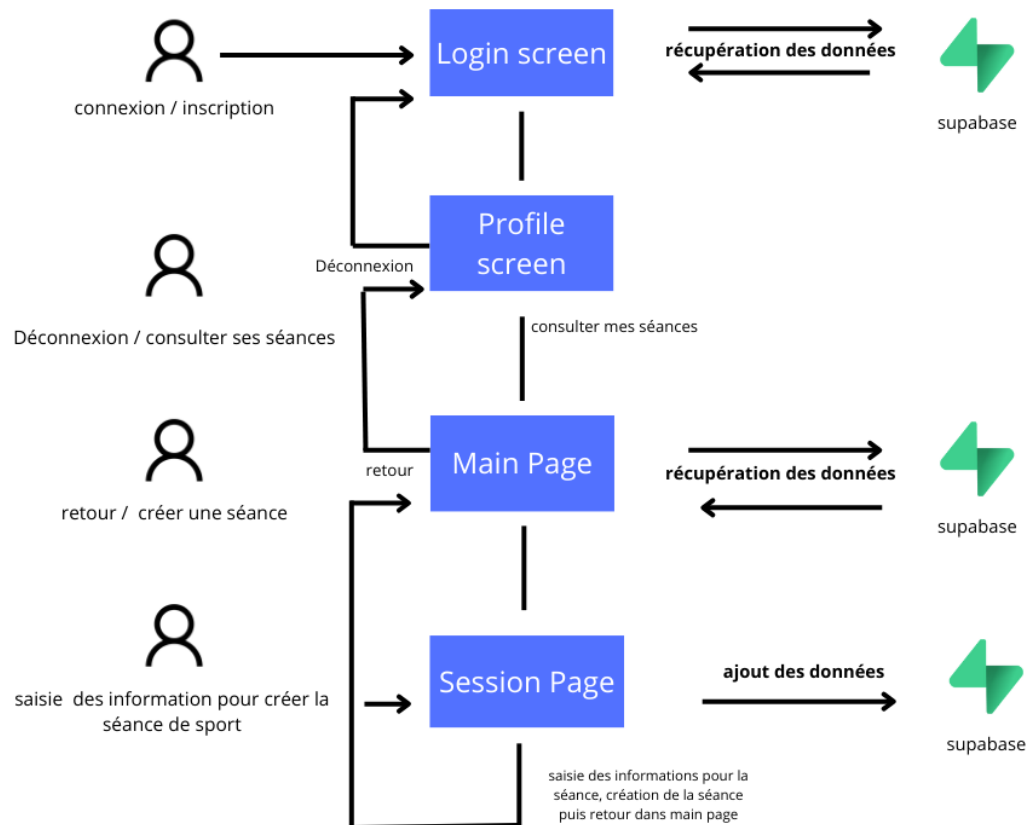
Lib :

- CreateSessionPage.dart : ce fichier dart est un screen qui permet à l'utilisateur de saisir toutes les informations Nécessaire pour créer une séance. C'est ici qu'on créer une séance qui s'ajoute dans le home_page.dart. C'est à partir de ce fichier qu'on va appeler et faire des Requêtes vers l'API ninjas, pour nous retourner les Information dont on a besoin.
- Home_page.dart : à partir d'ici on peut visualiser toutes ses séances. De plus on peut cliquer sur le bouton « plus » qui nous emmènera vers le screen de création de séances.
- Login_screen.dart : Après le lancement de l'application, l'utilisateur doit se connecter. Ce fichier sert à afficher le formulaire de connexion. A partir d'ici, on peut créer un compte et se connecter. Ici on se connecte à la base de données supabase où sont stocker les données.
- Main.dart : ici on initialise, et on appelle le login screen Pour poursuivre l'utilisation de l'application.
- Profile_screen.dart : profil de l'utilisateur, avec son ID et son image. A partir d'ici, il peut soit se déconnecter ou aller consulter toutes ces séances.



L'architecture de notre Application :

Architecture FitPlanner



Guide utilisateur :

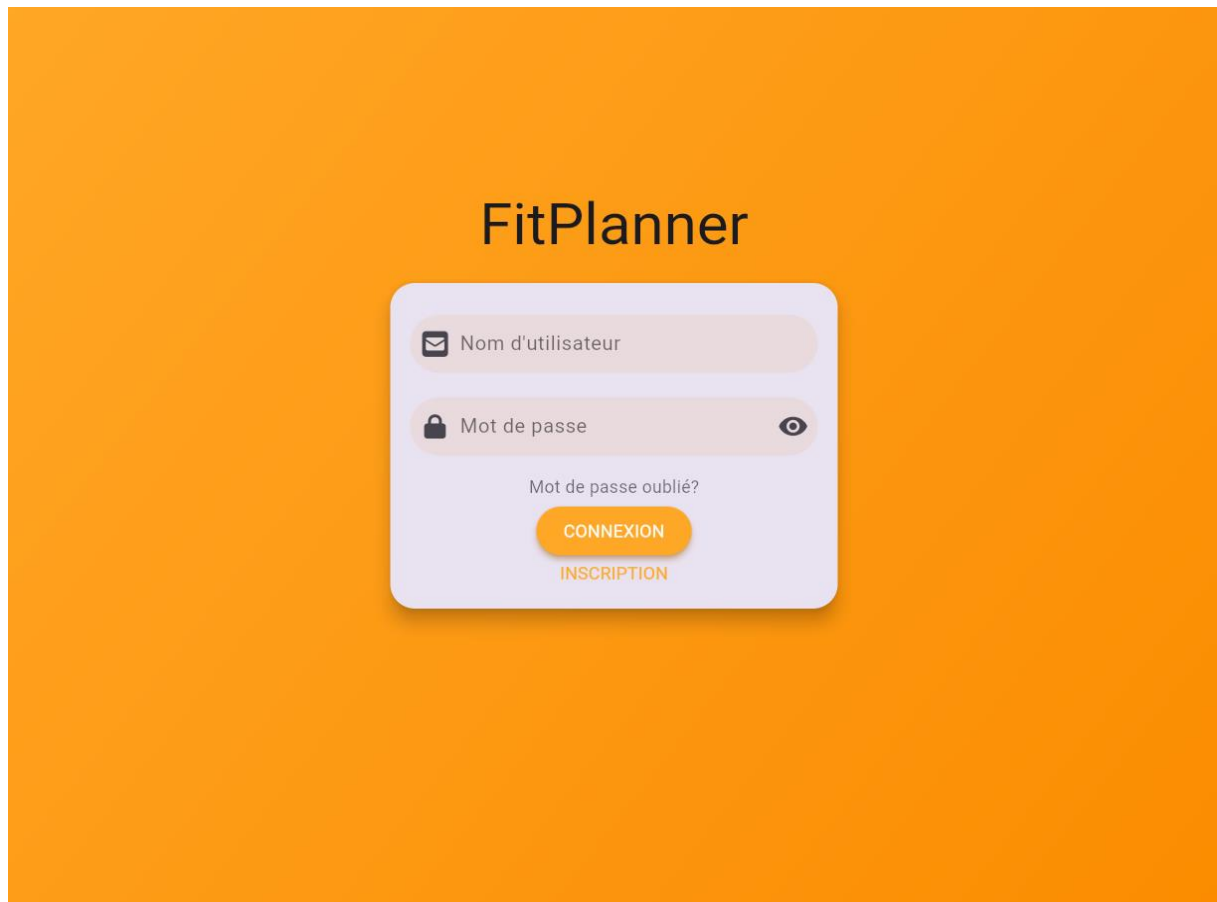


Fig 1 – Connexion ou Inscription

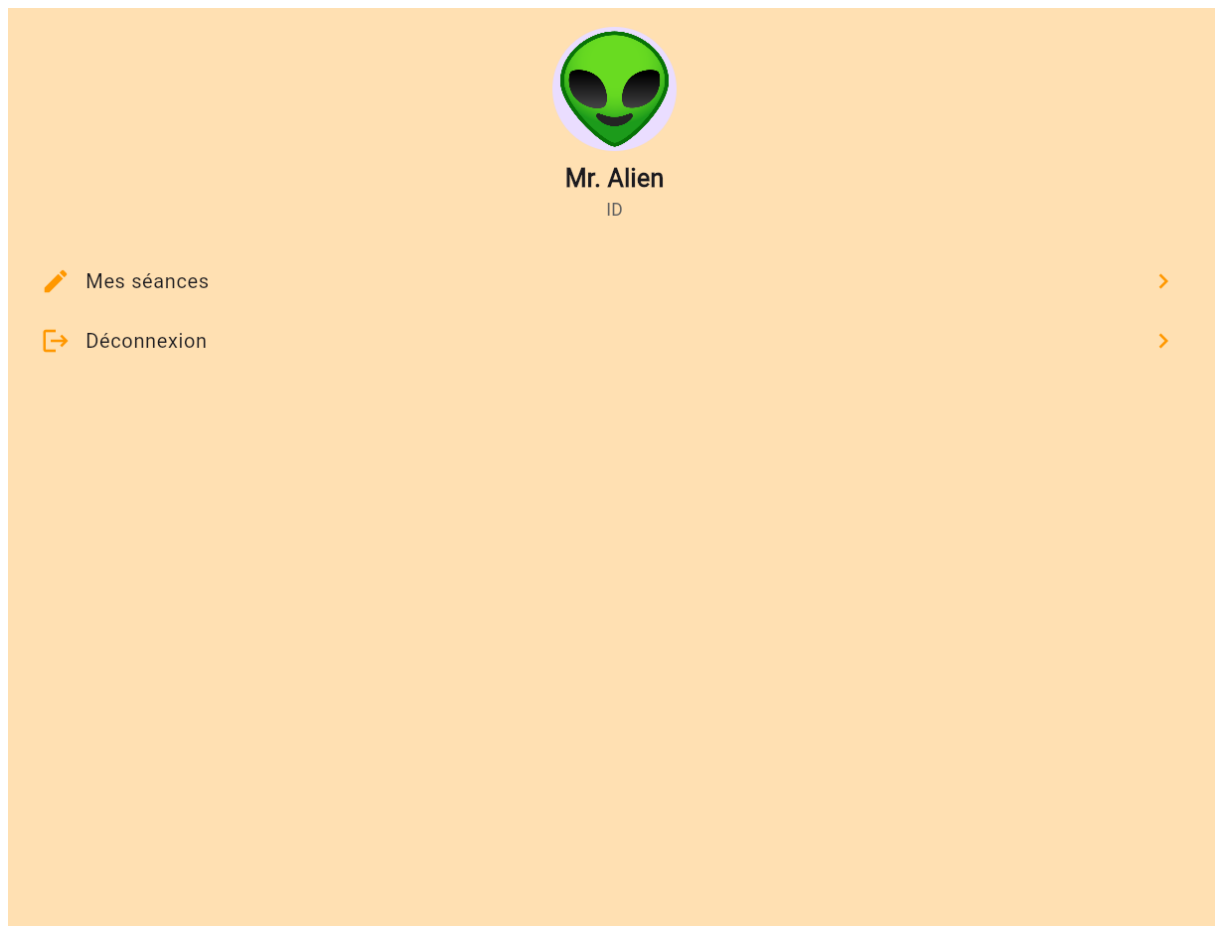


Fig 2 – Voici à quoi ressemble le profile screen quand un utilisateur se connecte. A partir d’ici on peut soit se déconnecter ou consulter ses séances.



Fig 3 – La page pour consulter ses séances, retourner au profile et ajouter une nouvelle séance

←

Créer une séance de sport

Nom de la séance

Durée (minutes)

Date de la séance

Aucune date sélectionnée

Description

Choose a muscle

Choose an exercise

+ Ajouter un exercice à la séance

Submit

Fig 4 – La page créer une séance : saisie du nom de la séance, la durée en minutes, date de la séance, la description, saisie du muscle et de l'exercice associé.

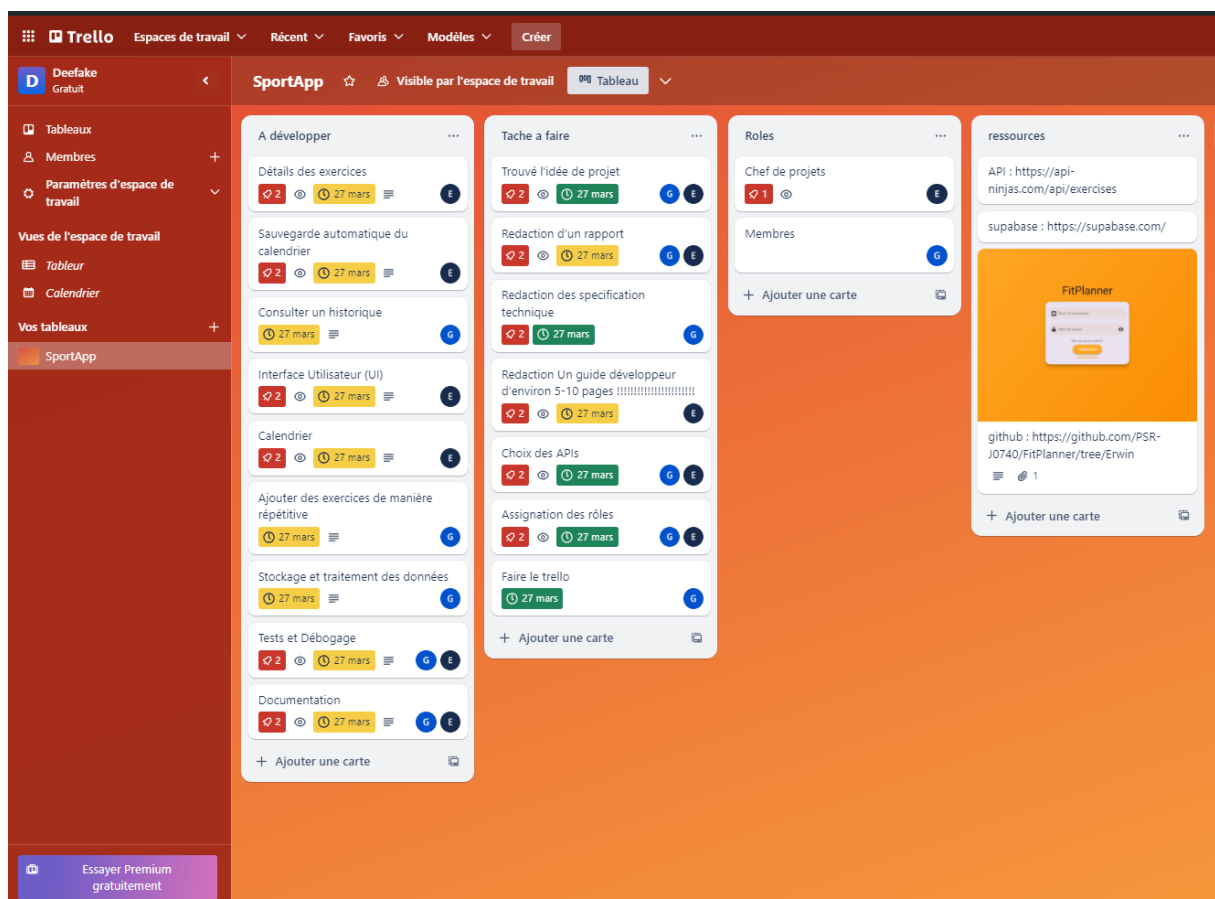
Gestion de projet

Nos outils de gestion de projet :

On a décidé de travailler avec l'outil de gestion de projet : Trello.

Nous avons pu s'organiser au sein de notre groupe et répartir toutes les tâches à réaliser, grâce à la création de différents tableaux :

- A développer : Ici nous avons toutes les tâches liées au développement de l'application, c'est plutôt une partie très technique
- Tâches à faire : Dans ce tableau nous avons toutes les tâches liées à la gestion de projet
- Rôles : Ici ont réparti les rôles de chacun
- Ressources : Tout les liens et ressources relié au projet



Nous avons utilisé GitHub pour la gestion de notre code. Chacun dispose de ça branche pour suivre l'avancement et les versions du code.

The screenshot shows the GitHub repository page for 'FitPlanner'. The repository is private and has 1 watch, 0 forks, and 0 stars. A notification at the top indicates that 'Erwin' had recent pushes 14 minutes ago, with a 'Compare & pull request' button. The repository has 3 branches (main, Erwin, Gwenole) and 0 tags. The 'main' branch is selected. The file list shows various Dart files and a README.md, with commit times ranging from 'last week' to '20 hours ago'. The right sidebar contains sections for 'About', 'Releases', 'Packages', 'Contributors' (listing PSR-J0740 and ErD000), and 'Languages' (showing 100.0% Dart).

File	Commit Time
code qui fonctionne	20 hours ago
Create README.md	last week
passwd, api key & api link for supabase	2 days ago
Add files via upload	last week
Add files via upload	last week
code qui fonctionne	20 hours ago
code qui fonctionne	20 hours ago
login and create account using a real database	3 days ago
Add files via upload	last week
same as the other two	3 days ago
code qui fonctionne	20 hours ago
Add files via upload	2 days ago
Add files via upload	3 days ago
Add files via upload	last week

Conclusion

Dans le cadre de notre projet FITPLANNER, nous avons entrepris la création d'une application dédiée à la planification et à l'organisation des séances de musculation. Ce projet, né de la volonté de fournir une solution numérique aux pratiquants de musculation, vise à remplacer les méthodes traditionnelles telles que l'usage du carnet papier ou la mémorisation des exercices. Notre objectif principal a été de développer une application qui offre une expérience utilisateur optimale, permettant une personnalisation poussée des séances de musculation, grâce à une interface intuitive et facile à utiliser.

Tout au long de ce projet, nous avons mis en œuvre une série de fonctionnalités clés telles que la création de compte utilisateur, la possibilité de concevoir des séances personnalisées incluant la sélection des muscles à travailler et des exercices associés, et la capacité à enregistrer et gérer plusieurs séances. Pour mener à bien ce projet, nous avons dû surmonter diverses contraintes techniques, notamment l'utilisation de Flutter pour le développement de l'application, l'intégration d'APIs, et la garantie d'une qualité de code élevée, avec pour objectif de produire un code structuré, commenté, et sans redondances.



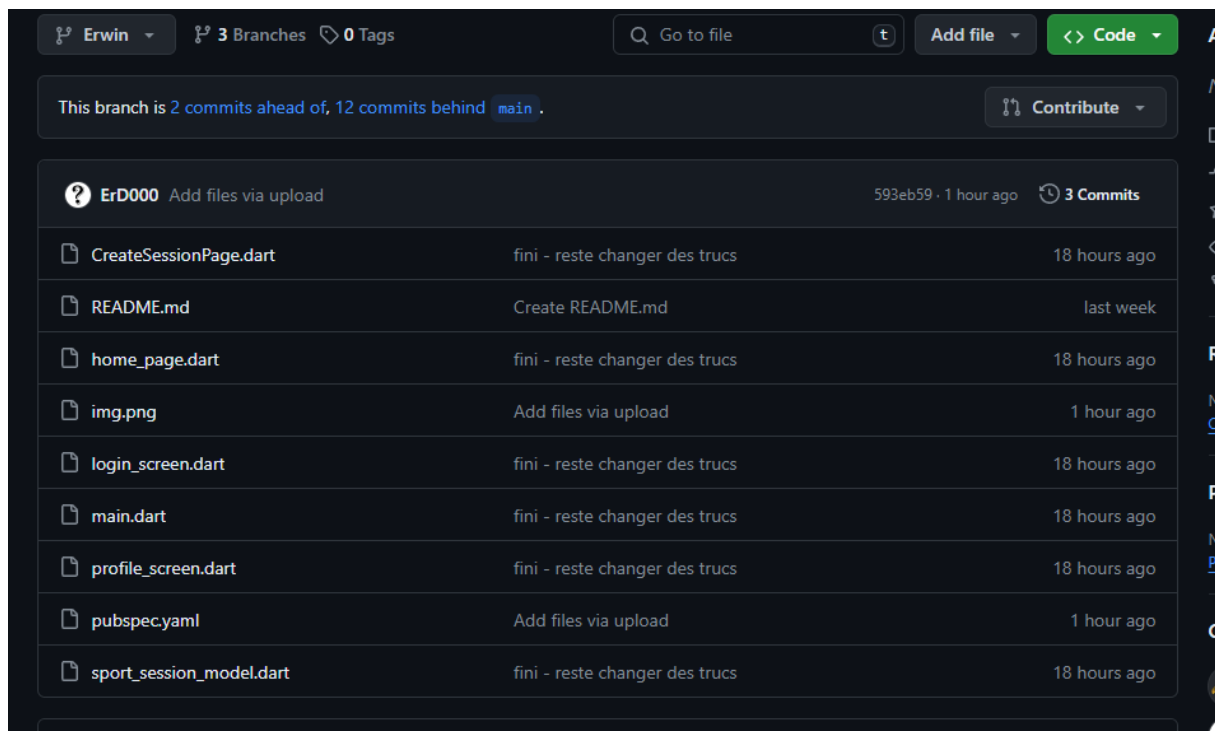
L'utilisation de technologies telles que Android Studio pour le développement, l'API NINJAS pour la récupération des données, et Supabase pour la gestion des données, a été cruciale pour la réussite de notre projet. Nous avons également adopté une approche méthodique pour la gestion de projet, en utilisant Trello pour l'organisation des tâches et GitHub pour la gestion du code, permettant ainsi une collaboration efficace au sein de notre équipe.

Code source

Notre dépôt :

Le code source de notre projet FitPlanner est hébergé sur GitHub, une plateforme de gestion de développement logiciel qui facilite la collaboration et le partage de code. Le dépôt contient l'intégralité du code source du projet, les fichiers de configuration, les documentations techniques, ainsi que les guides d'utilisation.

Le dépôt GitHub est accessible à l'adresse suivante : <https://github.com/PSR-J0740/FitPlanner.git>



Ressources et Références

Documentation officielle :

API Ninjas : <https://api-ninjas.com/api/exercises>

Supabase : <https://supabase.com/docs>

Flutter : <https://docs.flutter.dev/>

Tutoriels et Guides :

Flutter supabase : <https://www.youtube.com/watch?v=F2j6Q-4nLEE&list=PL5S4mPUpp4OtkMf5LNDLXdTcAp1niHjoL>

Flutter Json : <https://www.youtube.com/watch?v=Fo04xk9glFo>

Flutter Tutorial :
<https://www.youtube.com/watch?v=1ukSR1GRtMU&list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ>

Flutter Tutorial : https://www.youtube.com/watch?v=MQ1_wPyVLVc